REMARKS/ARGUMENTS

In the Office Action mailed on July 26, 2005, the Examiner rejected Clams 1-9, 12, 14-16, 18-20, 24, 26, 28, 30-33 and 35-51 under 35 U.S.C. §102(e) as anticipated by U.S. Patent No. 6,205,575 to Sherman ('575). Examiner also rejected claims 10, 17, 22, 29 and 34 as unpatentable over Sherman ('575) in view of Werner et al., and claims 11, 13, 23, 25 and 27 as unpatentable over ('575) in view of Ladkin et al.

Applicants amended independent Claims 1, 19, 20 and 39, and dependent claims 3-6 and 14-15 to clarify the subject matter of the application. In making these revisions care has been taken to ensure that no new matter has been added. All amendments are supported by the originally filed specification. Amendment to claim 1 is supported on page 6, lines 22 to 27 and page 7, lines 15 to 21. Amendment to Claim 14 is supported, for example, on page 12, line 15. Amendments to Claims 19 and 20 are supported on page 7, lines 18-21, and page 12, line15, respectively. Claim 39 as amended is supported by specification, page 12, line 15.

Applicants appreciate the time and consideration provided by the Examiner in reviewing this application, however, respectfully traverse the rejections of the claims at least for the following reasons.

Rejection under 35 U.S.C. §102(b)

Anticipation under 35 U.S.C. §102 requires that each and every claimed feature be disclosed by a single prior art reference. Applicants respectfully disagree that Sherman's '575 patent discloses the invention of the present application.

In order to provide a better understanding of the subject matter of the present application, Applicants present herewith an overview using a PowerPoint figure which shows the usage of a GUI of the system whose behavior is being specified. The Applicants believe that the presented drawing helps to indicate the difference between the method of the present application and the one shown in FIG. 8 of Sherman's '575 patent.

In Sherman's Fig. 8 the designer is using an interface of a standard CASE tool kind, which is a piece of software that lets the user draw scenarios manually by drawing objects, arrows, etc. These kinds of tools are well known in the industry, and have been around for 20 years.

In contrast, in the method of the present application, as illustrated by the attached PowerPoint figure, the user uses a Graphic User Interface (GUI) of a target application, for which the behavior is being specified. The user *does not* draw scenarios and is not concerned with the trivial notion of a tool that allows the user to draw rectangles and lines. Rather, the user uses the GUI as if it were a real application, plays with it, "telling" it how the user wants it to behave, which is done by actually showing the application — on the GUI — what the user wants it to do as a response to what user does, etc. (as described in detail in the present application). As this is going on, the GUI interacts with the Play-Engine, which, in turn, automatically generates the formally-defined scenarios in the form of Live Sequence Charts (LSCs).

In order to better understand the subject matter of the invention, the examiner is encouraged to go to the link for play engine demo - **http://www.wisdom.weizmann.ac.il/~playbook/demos.html**

The play-out mechanism disclosed in the present application greatly differs from prototype testing. In the prototype testing, one has to first *implement* the prototype. This includes programming the GUI and its behavior in some way, or programming the GUI and attaching it to a detailed design model that implements the previously specified system behavior. In any case, the behavior of each object (what we call the INTRA-object approach) has to be first specified in detail. And this can be done only *after* the design or implementation phase.

In contrast, in the method of the present application, the GUI is not attached in any way to the behavior and is unaware of it. More importantly, the behavior is specified using scenarios that describe the interactions between objects and changes in their properties (what we call the INTER-object approach). These scenarios are very high-level and correspond to multi-modal "stories" that can be specified during the requirements analysis phase. They include scenarios that CAN occur, ones that MAY occur, ones that MAY NOT occur, etc. And all these are executed by the play-out process taking these modalities into account.

Thus, the two main differences that distinguish the present application from the prior art are:

(i) in the prototype testing we test the design or implementation, - and in play-out we can execute and test the requirements themselves without the need to go through the design or coding phase, and

(ii) the behavior we execute in play-out is INTER-object and multi-modal, and does not necessarily contain behavior packaged for each object.

In view of the above, Applicants respectfully submit that the independent claims 1, 19, 20 and 39 as amended are novel in view of the prior art.

Rejection under 35 U.S.C.§103(a)

Claims 10, 17, 22, 29 and 34 rejected under 35 U.S.C. §103(a) as unpatentable over Sherman ('575) in view of Werner et al., and claims 11, 13, 23, 25 and 27 as unpatentable over ('575) in view of Ladkin et al.

Since all rejections under 35 U.S.C. §103(a) are based primarily on Sherman's '575, the arguments presented above are also applicable to the §103 rejection. As stated in MPEP 2142, in order to establish a prima facie case of obviousness, the references must teach or suggest all the claimed limitations. Applicants maintain that dependent Claims 10, 11, 13, 17, 22, 23, 25, 27, 29 and 34 should be deemed novel and non obvious over the cited references, *inter alia*, for the reasons discussed with reference to their respective independent base claims.

Therefore, the cited prior art references, alone and in combination, clearly teach away from the present application.

Applicants respectfully submit that all the pending claims as originally presented and amended by this response are allowable, and the application is now in condition for allowance, which allowance is earnestly solicited.

The Commissioner is hereby authorized to charge any fees, which may be required in connection with this correspondence, to Deposit Account No. 06-1135.

Respectfully submitted,

FITCH, EVEN, TABIN & FLANNERY

By: Kenneth H Sampl
Kenneth H. Samples
Registration No. 25,747

Date: 12/14/05

120 South LaSalle Street
Suite 1600
Chicago, IL 60603-3406
Telephone: (312) 577-7000
Facsimile: (312) 577-7007

Play-Engine Demos

David Harel & Rami Marelly

The aim of this web page is to give access to some simple and short movie clips that demonstrate some of the basic Play-Engine capabilities. All movies show recorded sessions of the tool, and they have not been edited. Although they were made with an earlier version of the Play-Engine than the one you have, the differences are minor.

Note: All videos are copyright 2003, Yeda Research and Development Company Ltd., the commercial branch of the Weizmann Institute of Science. You are welcome to view and save any of these movies, or to forward them in an e-mail if you would like, but each one only in its entirety and unedited. However, they may not be published or posted on a web site without the express, written consent of the authors.

Viewing Instructions: The movies are given both as .exe files and .avi files. The .exe files can be run on a Win32 platform with no need for a media player to be installed. In case the movie has to be shown on other platforms, the .avi format can be used. The demos are best viewed in 1280 x 1024 resolution (in lower resolution some parts of the screen may be hidden and can be viewed only using scrollbars). The movies can be stopped, paused and replayed by right-clicking anywhere on the screen.

Play-In

- Play-In Basics (PlayInBasics.exe)(PlayInBasics.avi)
  This movie demonstrates playing in a simple scenario.

- Symbolic Play-In (SymbolicPlayIn.exe)(SymbolicPlayIn.avi)
  This movie shows a symbolic scenario being played in.

Play-Out

- Simple Play-Out (PlayOutCalc.exe)(PlayOutCalc.avi)
  This movie shows a simple play-out session for a pocket calculator application.

Internal Objects

- Playing Internal Objects (InternalObjects.exe)(InternalObjects.avi)

This movie shows play-in and play-out with internal (GUI-less) objects.

Symbolic Instances

- **Playing with Symbolic Instances** (SymbolicInstances.exe)(SymbolicInstances.avi)
  This movie shows play-in and play-out in a phone network system with symbolic instances that represent classes rather than concrete objects.

- **The NetPhone Example** (NetPhone.exe)(NetPhone.avi)
  This movie shows a more elaborate play-out session with the phone network system, emphasizing the power of symbolic instances.

Time Constraints

- **Timing Constraints** (SimpleTime.exe)(SimpleTime.avi)
  This movie demonstrates play-in and play-out of simple timing constraints.

Smart Play-Out

- **Smart Play-Out Helps** (SmartHelps.exe)(SmartHelps.avi)
  This movie provides a simple demonstration of smart play-out (execution that is driven by model-checking techniques) and how it helps in computing a "good" superstep. First, it shows two LSCs that lead to two different super-steps by different interleavings. It then shows how the naive play-out algorithm finds a super-step in which one chart cannot be completed, and then how smart play-out finds the correct order of events, bringing both charts to completion.

- **Detecting Inconsistencies** (SpecInconsistency.exe)(SpecInconsistency.avi)
  This movie shows how model-checking techniques are utilized to find inconsistencies in LSC specifications. In this example, smart play-out is used to compute a super-step for a specification that contains two contradicting charts, and ends up deciding that there is no possible legal super-step .

- **Satisfying Existential LSCs** (SatisfyExistential.exe)(SatisfyExistential.avi)
  This movie shows how model-checking techniques are utilized to satisfy existential LSCs. We first show an existential LSC that serves as a test. The smart play-out module is called to try to satisfy it, and in response returns a sequence of events that satisfies the chart without violating any universal chart. This sequence of events is fed into the Play-Engine automatically and is executed.

Back to main page